

2022テスト設計コンテスト 発表資料

チーム： ジョゼ

株式会社HBA



- テスト設計の背景
- 要求分析・アーキテクチャ設計
- テスト詳細設計
- スコープの決定方法
- テスト実装
- 非機能要件について
- まとめ

テストの背景

システム開発：ASTER社

テスト開発：ジョゼ社



ASTER社からの依頼はテスト管理ツールの「システムテスト設計」

- 提供されるもの：ユーザーマニュアルと提案書、実際に動作するソフトウェア
- ツールは既に多くの企業で導入されており、最近是他社製のテスト管理ツールとの差別化のために機能アップデートの機会が増えている
- テスターはテスト設計会社と同じ会社に所属しており、評価に慣れたチームで構成されている

**頻度が増える機能アップデートに対応した
テスト設計を検討する**

1. テスト設計の背景 保証する内容

ジョゼの保証前提

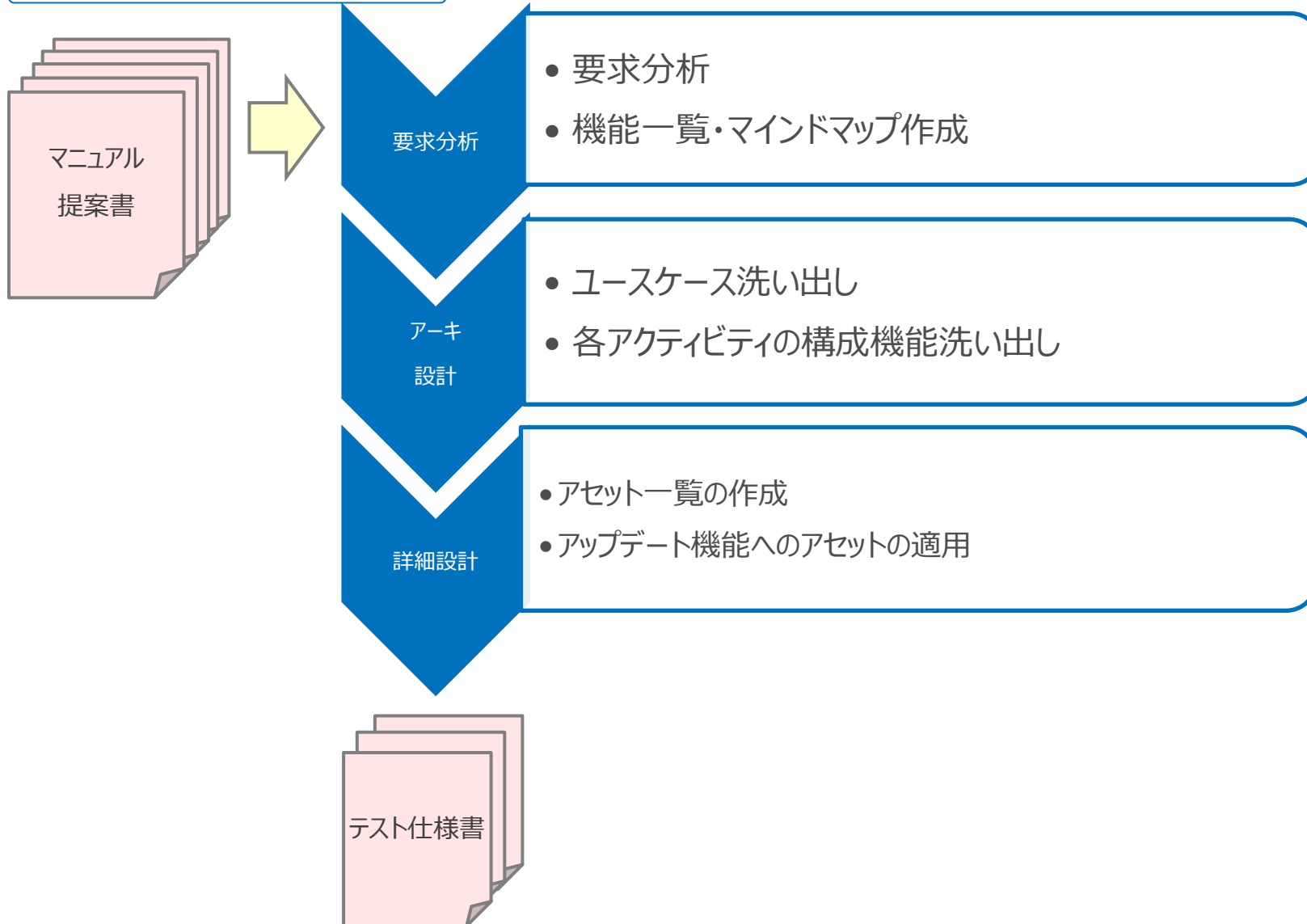
- 既存機能に対する評価は実施済み
- 評価対象は基本的にはアップデートの機能及びその影響範囲に対して行う

テストアーキの検討

- システムの全体像を機能一覧・マインドマップで表現
- アップデート機能に対し、ユースケースを検討
- ユースケースの全フローの各アクティビティに対し、構成する機能を洗い出すことで要求を具体化していく

1.全体のプロセス

テスト設計の進め方



2. 要求分析・アーキ設計

Input

ユーザーマニュアル 提案書 現行ソフト

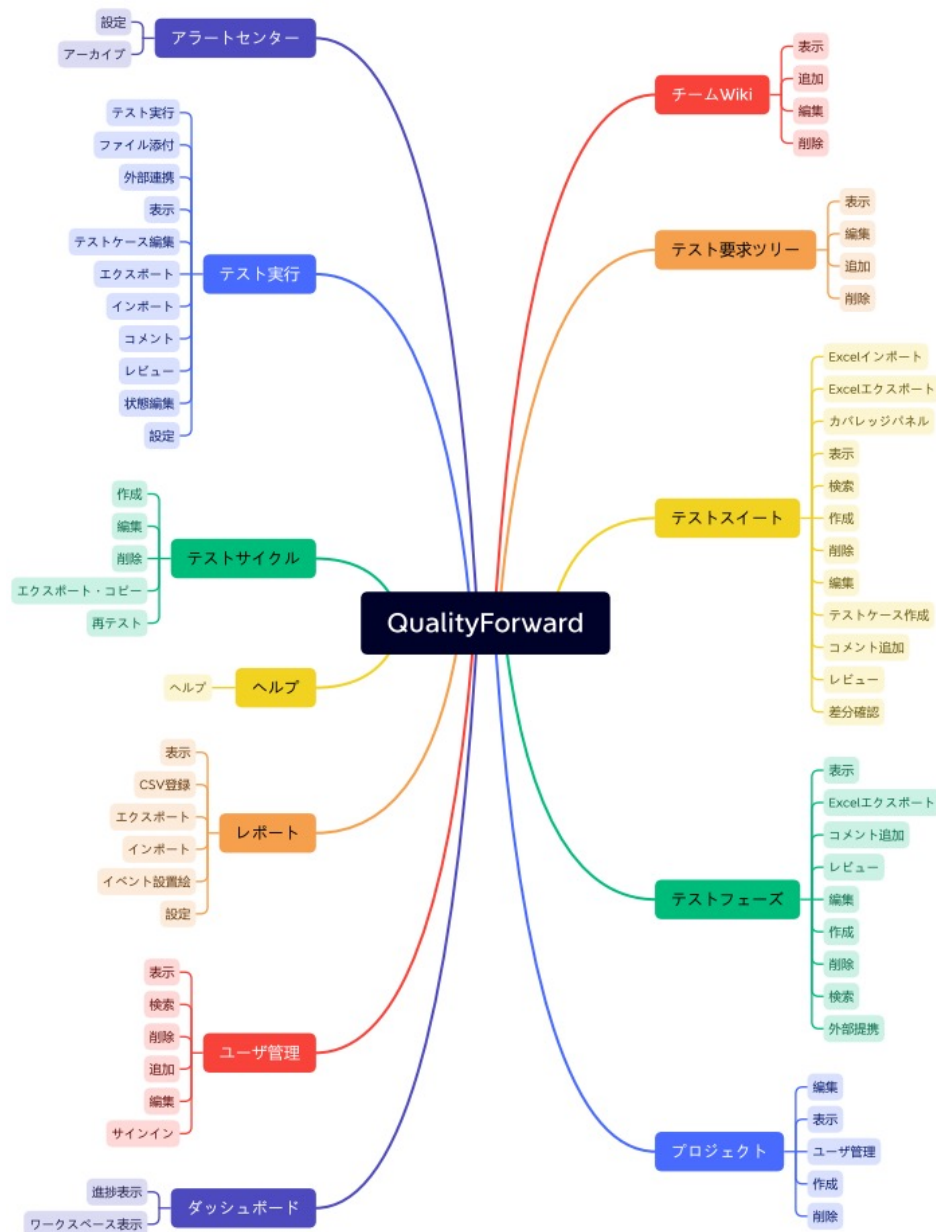
Point

ユーザーマニュアルを及びシステム動作を元に、機能洗い出しを行う

機能1	機能分類2	機能分類3	機能分類4	機能分類5
テスト実行	レビュー	レビュー	依頼	テキスト入力
テスト実行	レビュー	レビュー	差し戻し	テキスト入力
テスト実行	レビュー	レビュー	完了	テキスト入力
テスト実行	状態編集	状態編集	-	-
テスト実行	設定	設定	-	-
レポート	表示	プロジェクト全体	収束曲線	-
レポート	表示	プロジェクト全体	カバレッジパネル	-
レポート	表示	フェーズごと	収束曲線	-
レポート	表示	フェーズごと	カバレッジパネル	-
レポート	表示	フェーズ一覧	-	-
レポート	表示	インシデント一覧	-	-
レポート	表示	チャート表示	バグ優先度	-
レポート	表示	チャート表示	ステータス	-

機能一覧

マインドマップ

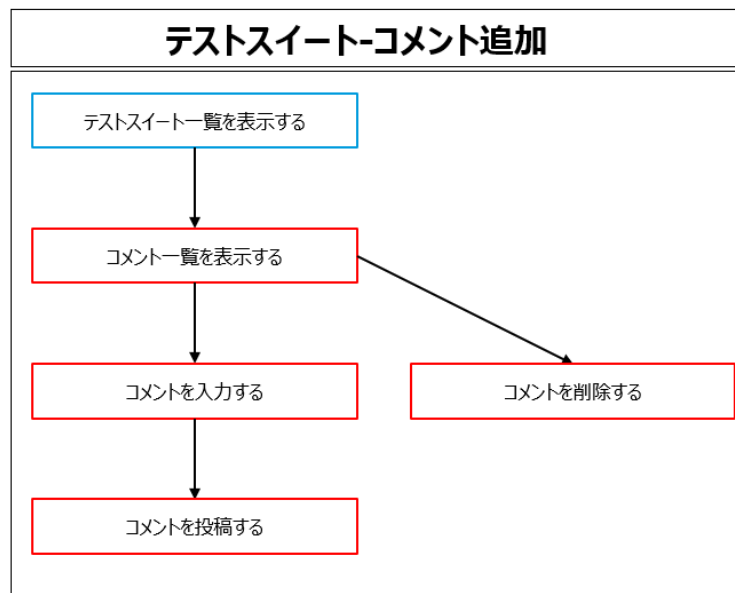


Input

機能アップデートの要求概要

Point

要求インプットの抽象度が高い場合に、まずユースケースを明らかにすることでアップデート機能を実現するアクティビティを明確化する



ユースケースフロー（ユースケースコンテナ）

2. 要求分析・アーキ設計

Input

ユースケースフロー（ユースケースコンテナ）

Point

ユースケースフローの各アクティビティに対しできること（＝機能）を洗い出す

テストスイートにコメントを投稿する

アクティビティ	時系列			
	テストスイート一覧を表示する	コメント一覧を表示する	コメントを入力する	コメントを投稿する
できること	テストスイート一覧が表示できる テストスイート一覧にコメントボタンが表示できる コメントボタンにコメント数が表示できる	コメントが表示できる コメント入力のテキストボックスが表示できる コメントの投稿ボタンが表示できる コメントの日時が表示できる コメントの削除アイコンが表示できる（自分のコメントの場合） コメント入力者の名前が表示できる コメント入力者のアイコンが表示できる	テキストボックスに文字が入力できる	投稿したコメントがコメント一覧に表示できる テキストボックスを空欄にする 投稿したコメントの日時が表示できる 投稿したコメントの削除アイコンが表示できる 投稿したコメント入力者の名前が表示できる 投稿したコメント入力者のアイコンが表示できる

Point

アセットとは、テストの観点をまとめたもの

ID	アセット	水準
1	BTS連携先	Redmine,JIRA
2	CSV拡張子	csv, csv以外
3	Excel拡張子	xlsx, xlsxm, xlsb
4	Wiki記法	テキスト入力, Markdown
5	アイコン設定	デフォルト, ユーザ設定済み
6	アカウントユーザのロック状態	ロック、ロック解除
7	アラートの条件設定状態	設定あり、設定無し
8	アラート状態	アクティブ、アーカイブ、削除
9	ステータス	編集集中、利用可、利用不可
10	テストアイテム数(サイクル、フェーズ、タグ)	0, 1, 9999, 10000
11	テストサイクル種別	実施中、レビュー待ち、完了
12	テストサイクル状態	未設定、テスト中、レビュー待ち、完了
13	テストフェーズ種別	実施中、完了
14	テストフェーズ状態	アクティブ、アーカイブ、削除
15	テスト結果種別	PASS, FAIL, SKIP, CUT, BLOCK, N/A, Q&A
16	テスト要求ツリーの階層	0, 1, 999
17	パスワード入力結果	一致、不一致
18	パスワード文字数	8文字以上、8文字未満
19	ファイルサイズ	0b, 1b, 1kb, 10mb以上
20	ファイル数	0, 1, 10000
21	フィルタ紐づけ数	0, 1, 100
22	プロジェクト状態	アクティブ、アーカイブ、削除
23	プロフィール画像	拡張子、サイズ
24	ユーザの権限	テナント管理者、プロジェクト管理者、一般ユーザ
25	ユーザ数	0, 1, 9999
26	ユーザ登録状態	未登録、登録ユーザ、削除済みユーザ

アセットの例

3. 詳細設計 アセットの作成

Point

アップデート機能に対してアセット紐づけを行う
(例：コメント一覧表示機能)

※既存のアセット一覧にない場合は新規追加を行う

ID	アセット	水準
1	BTS連携先	Redmine,JIRA
2	CSV拡張子	csv, csv以外
3	Excel拡張子	xlsx, xlsxm, xlsb
4	Wiki記法	テキスト入力, Markdown
5	アイコン設定	デフォルト, ユーザ設定済み
6	アカウントユーザのロック状態	ロック, ロック解除
7	アラートの条件設定状態	設定あり, 設定無し
8	アラート状態	アクティブ, アーカイブ, 削除
9	ステータス	編集済, 利用可, 利用不可
10	テストアイテム数(サイクル, フェーズ, タグ)	0, 1, 9999, 10000
11	テストサイクル種別	実施中, レビュー待ち, 完了
12	テストサイクル状態	未設定, テスト中, レビュー待ち, 完了
13	テストフェーズ種別	実施中, 完了
14	テストフェーズ状態	アクティブ, アーカイブ, 削除
15	テスト結果種別	PASS, FAIL, SKIP, CUT, BLOCK, N/A, Q&A
16	テスト要求ツリーの階層	0, 1, 999
17	パスワード入力結果	一致, 不一致
18	パスワード文字数	8文字以上, 8文字未満
19	ファイルサイズ	0b, 1b, 1kb, 10mb以上
20	ファイル数	0, 1, 10000
21	フィルタ紐づけ数	0, 1, 100
22	プロジェクト状態	アクティブ, アーカイブ, 削除
23	プロフィール画像	拡張子, サイズ
24	ユーザの権限	テナント管理者, プロジェクト管理者, 一般ユーザ
25	ユーザ数	0, 1, 9999
26	ユーザ登録状態	未登録, 登録ユーザ, 削除済みユーザ



New !

50

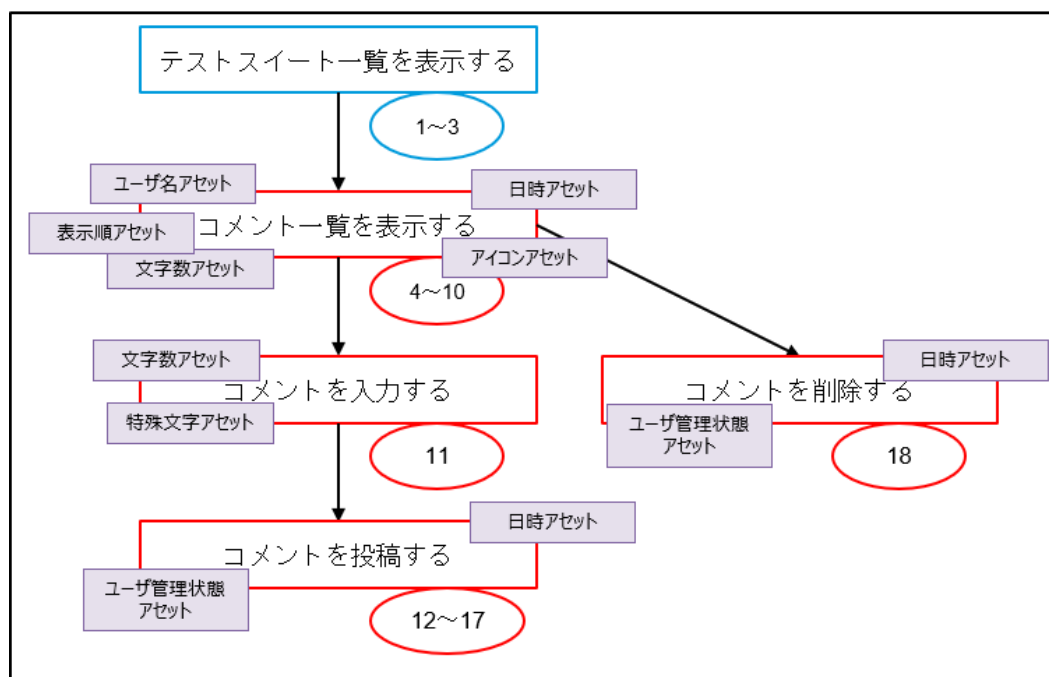
表示方法

アラート表示, ポップアップ表示, 通常表示

ID	アセット	水準
5	アイコン設定	デフォルト, ユーザ設定済み
26	ユーザ登録状態	未登録, 登録ユーザ, 削除済みユーザ
31	時間	0, 0, 1, 1, 9999
35	日付	TODAY, 1900/01/01 00:00:00, 9999/12/31 23:59:59,
37	表示順	降順, 昇順
39	文字種別	英字, 数字, ひらがな, 漢字, 記号, 絵文字, 特殊文字, 空欄, ドメイン有無
40	文字数	0, 1, 100, 10000

Point

アップデート機能に対しアセットを適用する



ユースケースコンテナ

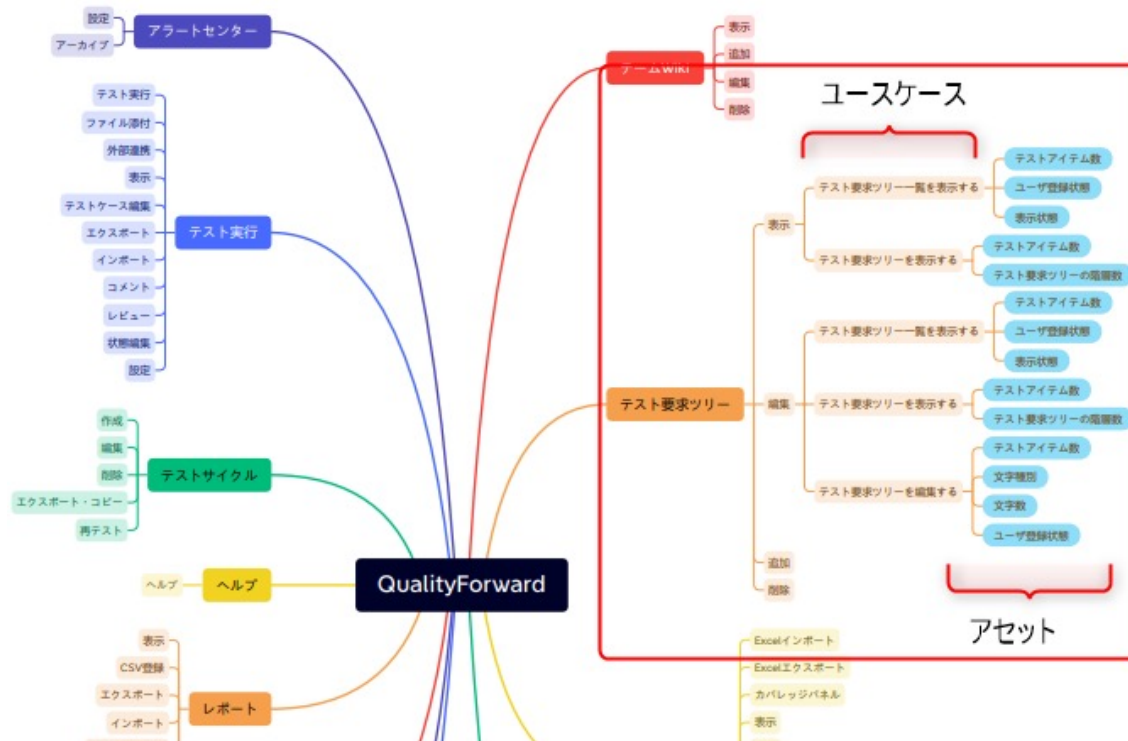
No	できること
1	テストスイート一覧が表示できる
2	テストスイート一覧にコメントボタンが表示できる
3	コメントボタンにコメント数が表示できる
4	コメントが表示できる
5	コメント入力のテキストボックスが表示できる
6	コメントの投稿ボタンが表示できる
7	コメントの日時が表示できる
8	コメントの削除アイコンが表示できる
9	コメント入力者の名前が表示できる
10	コメント入力者のアイコンが表示できる
11	テキストボックスに文字が入力できる
12	投稿したコメントがコメント一覧に表示できる
13	テキストボックスを空欄にする
14	投稿したコメントの日時が表示できる
15	投稿したコメントの削除アイコンが表示できる
16	投稿したコメント入力者の名前が表示できる
17	投稿したコメント入力者のアイコンが表示できる
18	自分のコメントがコメント一覧から削除できる

Point

★利点

- ・ 観点アセットをあらかじめ整理しておくことで、各機能・要求へのテスト観点を考える工程を効率化する
- ・ 観点漏れ分析等や問題の切り分けに有用
 - 各ステークホルダーに対する説明や合意形成に役立つ
- ・ 他システムにも応用することが可能：蓄積，洗練できる

3. 詳細設計 アセットの作成



【決勝の指令】

[テストアーキテクチャで改善すべき点を示したうえで見直す]

- ・ システムの全体像が見えづらい
- ・ 各設計プロセスの繋がりが見えない

への対応

全体図を明確化することでテストの対象やアセットの紐づけを考えやすくする

4. テスト範囲の決定

Point

- ・ 考え方 1

親子兄弟を対象とする

例：「配置」にアップデート

→その兄弟の機能「フィルタ，表示・非表示，履歴表示など」も評価の対象とする

機能1 ▼	機能分類2 ▼	機能分類3 ▼
テスト実行	表示	フィルタ
テスト実行	表示	配置
テスト実行	表示	表示・非表示
テスト実行	表示	設定表示
テスト実行	表示	関連ファイル表示
テスト実行	表示	実行時間表示
テスト実行	表示	履歴表示
テスト実行	テストケース編集	直接編集
テスト実行	テストケース編集	Excelインポート

4. テスト範囲の決定

Point

・考え方2

同じ機能分類名を対象とする

例：「テキスト入力」にアップデート

→テキスト入力が行われる機能に影響すると考え、それらの機能分類をテスト対象とする

機能1	機能分類2	機能分類3	機能分類4
プロジェクト	作成	基本設定	テキスト入力
プロジェクト	作成	ラベル設定	テキスト入力
プロジェクト	作成	結果補足設定	テキスト入力
テスト要求ツリー	編集	要素追加	テキスト入力
テスト要求ツリー	追加	基本設定	テキスト入力
テストスイート	検索	キーワード	テキスト入力
テストスイート	検索	タグ	テキスト入力
テストスイート	作成	名称設定	テキスト入力
テストスイート	作成	定義項目設定	テキスト入力
テストスイート	作成	自由項目設定	テキスト入力
テストスイート	作成	初期バージョン設定	テキスト入力
テストスイート	作成	新バージョン作成	テキスト入力
テストスイート	編集	設定変更	テキスト入力
テストスイート	編集	ヘッダ設定	テキスト入力
テストスイート	テストケース作成	オンライン編集	テキスト入力
テストスイート	コメント追加	追加	テキスト入力
テストスイート	レビュー	レビュー	テキスト入力
テストスイート	レビュー	レビュー	テキスト入力
テストフェーズ	作成	基本設定	テキスト入力
テストフェーズ	編集	基本設定	テキスト入力
テスト実行	テストケース編集	直接編集	テキスト入力
テスト実行	コメント	追加	テキスト入力
レポート	設定	ラベル設定	テキスト入力

4. テスト範囲の決定

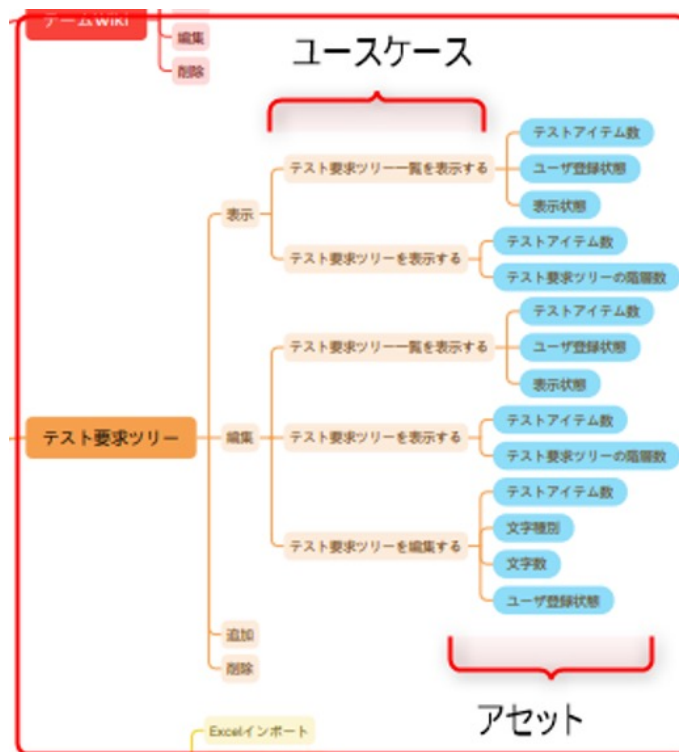
Point

・考え方3

同じアセットが紐づいているものを対象とする

例：ユーザ登録状態の判定部分にアップデートが入ったとする

→ユーザの登録状態を基準として動作する機能は影響を受け得るとして、
「ユーザ登録状態アセット」が紐づいている機能をテスト対象とする



Point

テスト観点：アセット一覧

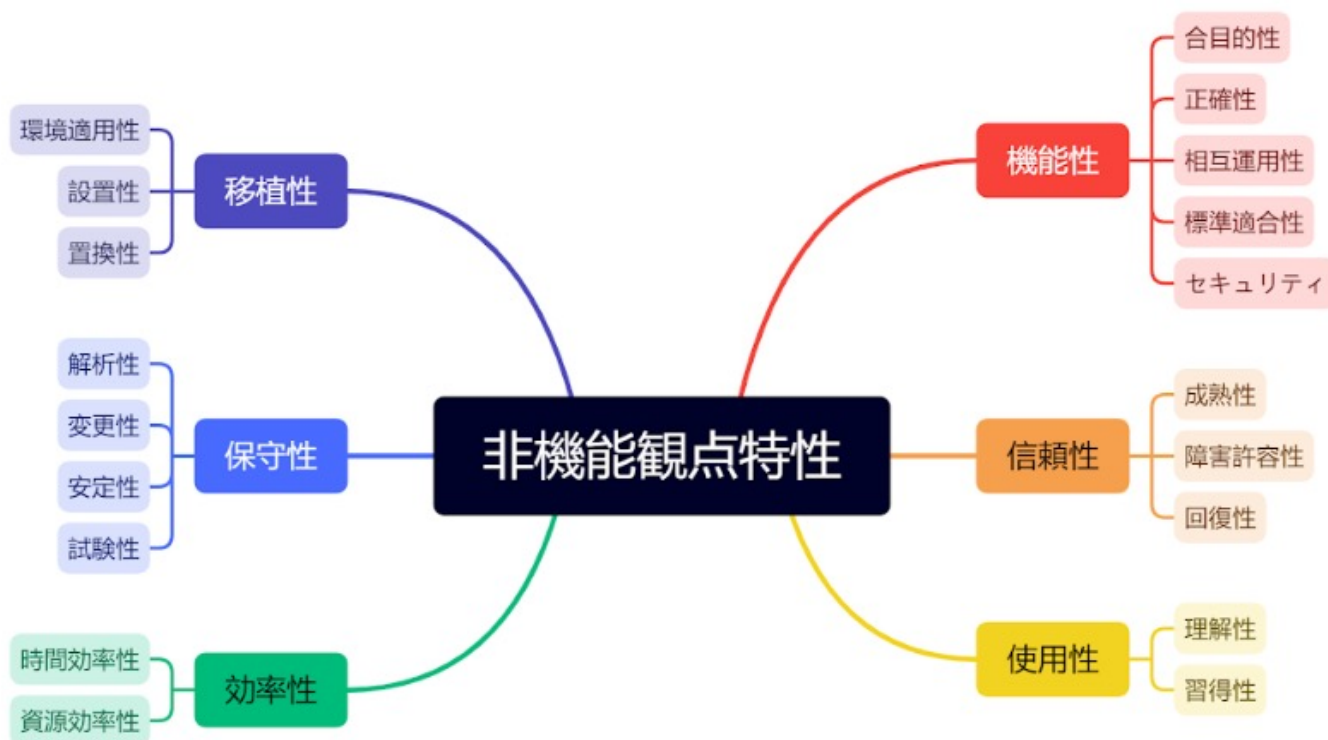
手順：ユースケースフロー図（コンテナ）

期待値：アセット内の要素

No	テストタイプ	機能分類23	手順(アクティビティ)	テストケース目的	観点アセット	期待値
1	機能テスト	テストスイート-コメント追加	①テストスイート一覧を表示する ②コメント一覧を表示する ③コメントを入力する ④コメントを投稿する	コメントが表示できる	文字数	②1文字のコメントが表示できること
2	機能テスト	テストスイート-コメント追加	①テストスイート一覧を表示する ②コメント一覧を表示する ③コメントを入力する ④コメントを投稿する	コメントが表示できる	文字数	②10000文字のコメントが表示できること
8	機能テスト	テストスイート-コメント追加	①テストスイート一覧を表示する ②コメント一覧を表示する ③コメントを削除する	自分のコメントがコメント一覧から削除できる	ユーザの権限	③一般ユーザの自分のコメントがコメント一覧から削除できること

Point

- 品質特性・副特性を元に関連する観点を洗い出す
→各観点に対し基準を設定し，それを非機能要求とする



6. 非機能要件のテスト観点

Point

- 品質特性・副特性を元に関連する観点を洗い出す
→各観点に対し基準を設定し，それを非機能要求とする

特性	副特性	観点	基準(例)
機能性	標準適合性	規格順守	x規格を順守していること
機能性	標準適合性	法規順守	x法を順守していること
機能性	セキュリティ	SQLインジェクション	エスケープ処理がなされていること
機能性	セキュリティ	バッファオーバーフロー	領域溢れを起こしていないこと
信頼性	成熟性	エラー発生率	99.99%(1年間で1時間程度)
信頼性	障害許容性	エラーケース	インターネット接続切断時にデータが破棄されないこと
信頼性	回復性	RPO	0秒(停止直前までのデータが必要)
信頼性	回復性	RTO	6時間
信頼性	回復性	RLO	50%のエンドユーザ数
使用性	理解性	マニュアル適切性	マニュアル通りの操作で機能を達成できること
使用性	習得性	マニュアル適切性	マニュアル通りの操作で機能を達成できること
効率性	時間効率性	応答時間	1秒
効率性	資源効率性	ハードウェアリソース使用率	10%
保守性	解析性	デバッグログ	デバッグログが記録されていること
保守性	解析性	エラーメッセージ	エラーメッセージが表示されること



No	テストタイプ	特性	副特性	テストケース目的	手順	期待値
1	非機能テスト	セキュリティ	SQLインジェクション	エスケープ処理がなされていること	文字入力画面でSQL Injectionを入力する	入力不可能なこと
2	非機能テスト	保守性	安定性	高負荷状態で各機能が実行できることを確認する	10000ユーザの同時アクセスを実施する	正常に機能が動作すること
3	非機能テスト	保守性	安定性	高負荷状態で各機能が実行できることを確認する	1時間あたり50000回のページ処理を実施する	正常に機能が動作すること
4	非機能テスト	保守性	安定性	連続運転中でも実行している機能の停止、遅延が発生しないことを確認する	1200時間連続でのシステム動作を実施する	正常に機能が動作すること

Point

★テスト設計の段階ではなく、実際に評価を運用する段階で活用

頻度：その欠陥が発生する頻度を一人の人間が一日操作したときに発生する割合

システム改修における重大度：その欠陥を改修する際に発生する遅延やその他コストで重みづけ

エンドユーザーにおける重大度：その欠陥がエンドユーザー側で露になった場合に影響に重みづけ

頻度		点数
IV	1人あたり 10回以上	4
III	1人あたり 1/10～1回	3
II	1人あたり 1/100～1/10回	2
I	1人あたり 1/1000回以下	1
システム改修における重大度		点数
IV	プロジェクトが失敗する程度の遅延、あるいはコストが発生する	3
III	当初想定の1.3倍以上のコストがかかる遅延等が発生する	2
II	コスト増加は多くはないが、1日以上遅延等が発生する	1
I	遅延・追加コストなく対応可能である	0
エンドユーザーにおける重大度		点数
III	顧客指摘によるサービス影響が大きく対処が必要なもの	5
II	サービス改善のために取り組むべきもの	3
I	ユーザビリティ改善や一部ユーザーの声による軽微なもの	1

7. リスク分析の活用

Point

対応コスト：頻度，システム改修における重大度，エンドユーザーにおける重大度3つの点数を合計し，費用の観点で重みづけしたもの

観点	リスク	点数化				対応コスト
		①頻度	②システム改修における重大度	③エンドユーザーにおける重大度	①+②+③	
数値演算のパフォーマンスに問題ないか	DB性能が低下し、サービスが停止状態に陥る	1	2	3	6	Ⅱ
観点	リスク	点数化				対応コスト
		①頻度	②システム改修における重大度	③エンドユーザーにおける重大度	①+②+③	
大容量のCSVファイル格納時も問題ないか	Web操作に遅延が発生しユーザーに不満が出てくる 同一サーバーに収納されている他の顧客の業務に影響が出てユーザークレームになる	3	2	3	8	Ⅲ

上記合計値を下記表の「基準値」に割り当てコストを算定する。			
対応コスト	基準値		対応方針
Ⅳ 時価 (例)情報漏洩など対応にあたる費用が見積れないもの	10	→	対応必須
Ⅲ ¥1,000万円以上	8~9	→	個別案件により検討
Ⅱ ¥101万円~¥1,000万円以下	3~7	→	個別案件により検討
Ⅰ ¥100万円以下	2	→	個別案件により検討

今回のテスト設計でできること

- ・機能一覧と機能構成図（マインドマップ）を作成しシステムの全体像を明らかに
 - ・ユースケースフローとユーザーストーリーマッピング的思考方を用いて要求を具体化
 - ・アセット一覧を作成してテスト観点・条件を共通化
 - ・マインドマップを用いてテスト範囲を定義
- ※リスク分析により評価運用を効率化

→ 以上により、テスト設計を効率化し、頻発する機能アップデートに対応する

「IT」で「幸せ」に挑む。

